# ClipFlip : Multi-view Clipart Design

I-Chao Shen[1]   Kuan-Hung Liu[1]   Li-Wen Su[1]   Yu-Ting Wu[1]   Bing-Yu Chen[2]

[1] {jdily, lkh94, susan31213, kevincosner}@cmlab.csie.ntu.edu.tw, National Taiwan University, Taiwan
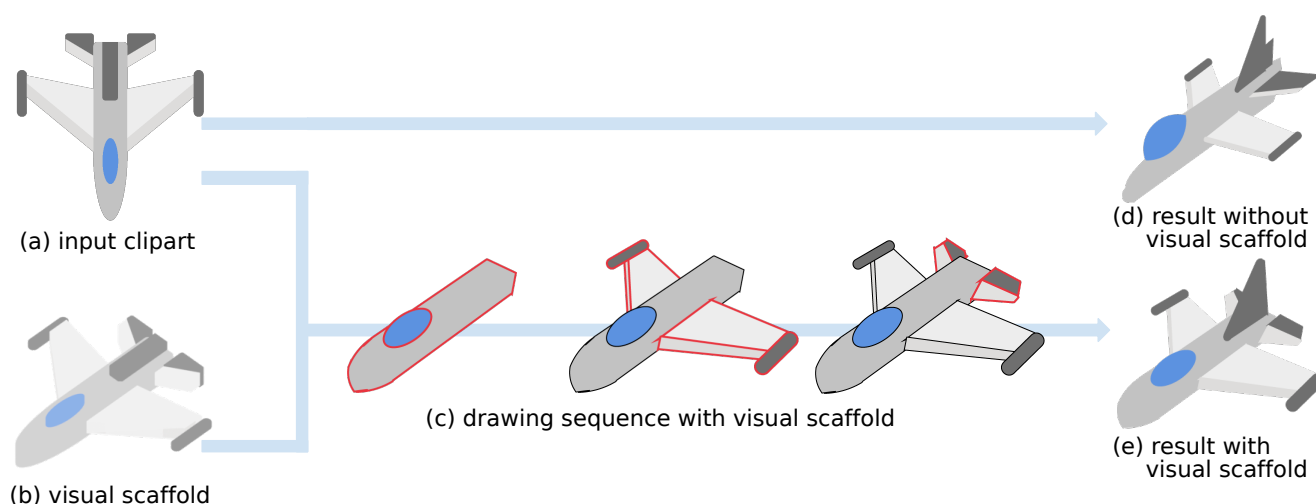[2] robin@ntu.edu.tw, National Taiwan University, Taiwan

Figure 1: Given the input clipart (a), Our system automatically generates a visual scaffold (b) by rendering it from a style consistent 3D shape generated by a user-assisted curve extrusion method. The user can design the clipart from the desired viewpoint by following the visual scaffold and draw each part step-by-step (c). We highlight the parts are being drawn at each step in red. Finally, the user can design better clipart (e) from the desired viewpoint in a shorter period than designing it without the provided visual scaffold (d).

**Abstract**

*We present an assistive system for clipart design by providing visual scaffolds from the unseen viewpoints. Inspired by the artists' creation process, our system constructs the visual scaffold by first synthesizing the reference 3D shape of the input clipart and rendering it from the desired viewpoint. The critical challenge of constructing this visual scaffold is to generate a reference 3D shape that matches the user's expectations in terms of object sizing and positioning while preserving the geometric style of the input clipart. To address this challenge, we propose a user-assisted curve extrusion method to obtain the reference 3D shape. We render the synthesized reference 3D shape with a consistent style into the visual scaffold. By following the generated visual scaffold, the users can efficiently design clipart with their desired viewpoints. The user study conducted by an intuitive user interface and our generated visual scaffold suggests that our system is especially useful for estimating the ratio and scale between object parts and can save on average 57% of drawing time.*

**Keywords:** vector graphics, clipart, multi-view, hci

**CCS Concepts**

• **Computing methodologies** → **Parametric curve and surface models;** Image processing; • **Human-centered computing** → *Graphical user interfaces;*

## 1. Introduction

Vector clipart is widely used in graphic design for compactly expressing concepts or illustrating objects in daily life. For example,

designers communicating their design ideas with others, or presenters illustrating concepts with cliparts. Whenever we need an icon in our slides or videos, we usually try to search it in online clipart repositories using our target keywords, such as a chair, a lamp, or an airplane. A common situation we encounter is that even if we are lucky enough to find clipart with satisfactory shapes and colors, sometimes it is designed from an undesired viewpoint. In the end, we often compromise with either a satisfactory appearance or a viewpoint.

When facing the above situation, one might consider manipulating the existing clipart to the desired viewpoint via editing tools. However, it is challenging to draw other views of an object with an only one available view due to the individual differences of their mental rotation ability [VK78]. We conducted interviews with two professional artists, and they tackle this task by first creating a rough 3D model based on the available view. The rough 3D model can then be used as the mental prior for designing clipart from other viewpoints. Unfortunately, this procedure requires professional training in 3D modeling and a good mental prior of the 3D object, which is almost impossible for general users.

In this paper, we propose a multi-view clipart assistive system to show unseen viewpoints of the clipart to support mental rotation. Our system follows the drawing-by-observation technique. Given the input clipart, our system's goal is to automatically synthesize other unseen viewpoints of the clipart object without changing the character of the clipart. Users can efficiently design high-quality clipart with their desired viewpoint by simply following the visual scaffold. Our system design simulates the creation process of professional artists. Instead of directly synthesizing unseen views from the 2D input, we first infer a reference 3D model based on the input clipart's viewpoint. We generate the reference 3D model with user-provided structural annotations that preserve the style of the input clipart. The visual scaffold is then generated by rendering the reference 3D model from the desired viewpoint. Finally, we also provide an intuitive user interface to assist users in their clipart creation.

The major contribution of our work is an algorithm for generating a style-consistent 3D model from the single-view clipart. Even though there are existing works focusing on novel view synthesis [ZTS*16, SHL*18, PYY*17, DSTB16, TDB16], there are two reasons that prevent us from directly applying these methods to serve our purpose. First, these methods require substantial training datasets to learn the categorical shape priors, but there is no existing huge clipart dataset that can facilitate such a learning process. Second, directly infer the novel view or 3D representation of the clipart object using the pre-trained models of the methods mentioned above usually lead to failure cases (Figure 2). The reason is that the object boundaries in clipart are mainly composed of low-degree curve types such as line and arc, not the cases in existing datasets. Due to the very different geometric styles, the predicted 3D geometry generated by previous methods would not be style-consistent with the input clipart. Our proposed user-assisted curve extrusion method addresses this problem by combining state-of-the-art 3D reconstruction and user-provided structural annotations to guide the curve extrusion method. The generated 3D shape is regularized and style-consistent with the style of the input clipart. Noted that we did not aim for proposing a novel deep learning architecture that
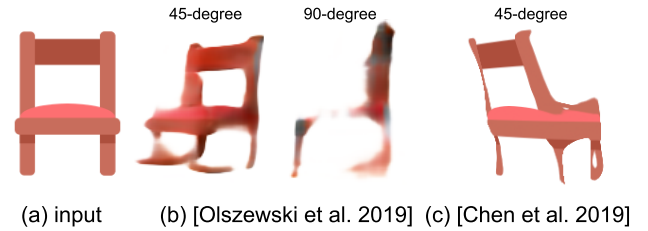


| 45-degree | 90-degree | 45-degree |
|-----------|-----------|-----------|

(a) input     (b) [Olszewski et al. 2019]   (c) [Chen et al. 2019]

Figure 2: Given the input clipat, both (b) Olszewski *et al.* [OTW*19] (TBN) and (c) Chen *et al.* [CSH19] fail to synthesize novel view of the input clipart. We use the pre-trained model for both methods provided on their webpages.

handles vector clipart directly. Instead, we leverage an existing 3D reconstruction method using a raster image and combine it with structural annotation to generate a regular 3D shape.

We conducted a user study with general users who did not have much drawing training or experience. The purpose is to evaluate how helpful our generated visual scaffold is for designing clipart from an unseen viewpoint. To conduct this user study, we designed a complete system for assisting general users in multi-view clipart creation. In our intuitive user interface, we show two display areas side-by-side: the *input reference area* and the *drawing area*. In the input reference area, we show the input clipart to the user, as most users will keep referring to the input viewpoint while drawing the novel viewpoint. In the drawing area, the user can use both curve tools and shape tools to design the novel view clipart. The visual scaffold we synthesized lies under the drawing area. With our system, most of the users could design better clipart from a novel viewpoint with a regularized visual scaffold in a small fraction of the original time. Furthermore, most participants agree that our visual scaffold significantly reduces their cognitive workload when designing clipart from a new viewpoint.

## 2. Related Work

### 2.1. Novel-view synthesis

To synthesize photo-realistic novel view images, most traditional methods [PZ17, SCD*06, KLS*13, DTM96] take multi-view images as input and infer the 3D representations explicitly to facilitate novel view rendering. On the other hand, recent learning-based methods [ZTS*16, SHL*18, PYY*17, DSTB16, TDB16] provide the ability to render novel-view images using only one single image. Some methods [DSTB16,TDB16] directly synthesize pixels of the desired viewpoint, while the other methods [ZTS*16,PYY*17, SHL*18] instead estimate the flow map from the input view to the desired viewpoint. Some methods [KLA19, NPLT*19] disentangled the image factors such as viewpoint, shape, and appearance from the source image and used these factors for facilitating view synthesis. There are also learning-based methods focusing on generating better data representation for synthesizing novel views from a single image, including voxel and volume-based representation [TZEM17, CXG*16, NMOG19], point cloud [FSG17, MWYG20], mesh [GFK*18,DN19,LLCL19], depth and/or normal maps [LGK*17,LPL*18], and parametric surface [SBS19].

The goal of our system is similar to novel view synthesis from a single-view image. However, unlike previous methods that focus on photo-realistic raster images and have many training data available, there is no existing clipart dataset (neither raster-based nor vector-based) that can facilitate such a learning-based approach. Meanwhile, Lopes *et al.* [LHES19a] proposed a generative model for font synthesis with a deep learning framework handles vector format. However, we can not apply this method directly for our application because their model did not handle (i) the color, (ii) shape category, and (iii) the viewpoint information. Hence, we combine a learning-based approach on raster image with user-provided structural annotation and provide an intuitive interface to aid the creative process.

### 2.2. Clipart synthesis

Vector clipart can be synthesized by vectorizing existing raster images or designing from scratch. Commercial products [Ado20, Vec20] provide robust image vectorization that simultaneously tackles both image segmentation and curve (segment boundary) fitting problem. However, the clipart style usually differs from the natural image (e.g., flat shadings and rounded shapes), which makes the vectorized natural image not suitable for synthesizing clipart. Some previous works have focused on vectorizing pixel arts [Ste03, KL11, HDS*18]. Kopf and Lischinski [KL11] proposed a dedicated method that focuses on resolving the topological ambiguities for pixel arts vectorization. Hoshyari *et al.* [HDS*18] used human perceptual cues to generate boundary vectorization that can better match viewers' expectations. Liu *et al.* [LALR16] designed an interactive system to synthesize novel clipart by remixing existing clipart in a large repository.

### 2.3. Assisting authoring tools

Content authoring is a fundamental problem in Computer Graphics. Many previous works focus on assisting the authoring workflow. Among them, many works utilize the personal editing histories to assist 2D sketch [XCW14], 3D shape sculpturing [PXW18], and viewpoint selection [CGW*14]. Besides the history-related assists, Xing *et al.* [XKG*16] utilizes the energy strokes from professional artists to facilitate the authoring process of 2D animation. Lee *et al.* [LZC11] proposed a guidance system that dynamically updates a shadow image underlying the user's strokes while the user is drawing. Hennessey *et al.* [HLW*17] automatically generates a step-by-step tutorial for drawing 3D models. Ryan *et al.* [SIJ*07, SKSK09] provide construction lines and primitives as visual scaffolds for better drawing 3D objects. Our system shares the same spirit and provides a visual scaffold to support users for designing clipart. The major difference is that we provide the accurate rendered image as a visual scaffold instead of hints such step-by-step as construction lines. According to the feedback from our study, the visual scaffold of the final rendered image can provide an overview of the shape in the beginning so that they can better arrange the order of drawing.

### 2.4. Geometric stylization

Creating 3D shapes with different styles of geometric has drawn a lot of attentions including Japanese manga style [SLHC12], Lego brick style [LYH*15], abstraction style [MZL*09, YK12], and cubic style [lJ19]. Lun *et al.* [LKS15, LKWS16] analyze the style similarities between shapes, and transfer different styles between shapes. Unlike the previous works that manipulate the input 3D shapes using mesh deformation techniques, our goal is to synthesize the 3D shape from the input clipart. We combine the single-view 3D mesh reconstruction method and user's structural annotations to guide the curve extrusion process and synthesize reference 3D shape that matches the input clipart in both geometric and appearance styles.

## 3. Visual Scaffold Synthesis

Our system includes two major components: (i) visual scaffold synthesis, which synthesizes the input clipart under a desired viewpoint, and (ii) the drawing user interface, which displays the synthesized visual scaffold beneath the user's drawing canvas to aid the drawing process. Given the input clipart consists of multiple closed paths $S = \{C_0, C_1, ..C_{n-1}\}$ under input viewpoint $\theta^i$, the goal of this step is to synthesize the visual scaffold that aids the users in designing the clipart from an unseen viewpoint $\theta^u$ efficiently. To achieve this goal, we design our method by addressing the following three aspects:

1. the shape in the visual scaffold image has to match the user's imagination for the input clipart from viewpoint $\theta^u$.
2. the appearance style of the visual scaffold has to match the input clipart.
3. the geometric style of the shape in the visual scaffold has to match the input clipart.

If the shape in the visual scaffold conflicts with the user's imagination (i.e., violate (1)), it will hinder the creative process instead of providing useful aids. In the meanwhile, if the geometric and appearance style did not match the input clipart (i.e., violate (2) and (3)), the clipart created by the user will not be depicted as clipart. To address the above observations, we propose a user-assisted curve extrusion method to generate a reference 3D shape from the input clipart. The user is allowed to provide structural annotations on the input clipart to indicate the 3D structure information. As illustrated in Figure 3, our method generates a reference 3D model $M$ by following the guidance of both user's structural annotations and the guiding mesh $M^G$, which is reconstructed by using a single-view mesh reconstruction method based on the input clipart $S$. We search for optimal thickness and transformations of extrusion to simultaneously match users' structural annotations, while best interpreting the guiding shape $M^G$. Noted that we use the rasterized curves of the input clipart $S$ for inferring the guiding mesh. The main reason for choosing the raster-based method is because unlike the mature architecture for raster image, such as CNN (convolutional neural network), there is no mature learning-based method that handles the vector clipart well. Moreover, this paper aims not to propose a novel deep learning architecture that directly handles vector clipart. Even though there exists relevant work [LHES19b] that handles vector format data using a deep generative model, however, their model did not consider (i) the color, (ii) shape category, and (iii) the viewpoint information. These pieces of information are vital to our application. So we choose to leverage more mature methods
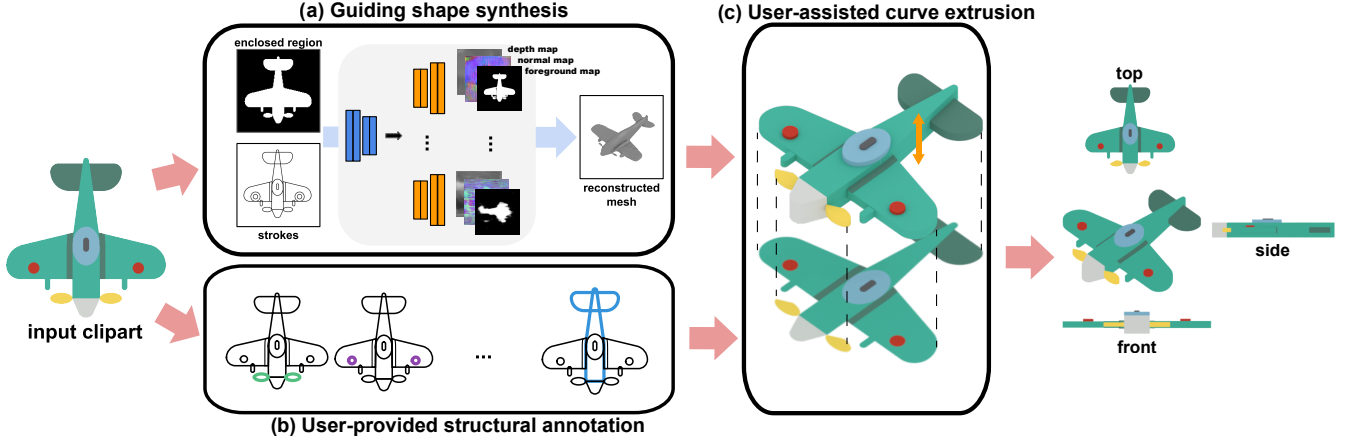
Figure 3: Given the input clipart, our method (a) use a single-view 3D reconstruction method to synthesize a guiding shape. (b) The user can annotate the input clipart to provide the structural information. (c) Our method extrude the curves in the input clipart adaptively by leveraging the predicted guiding shape and the user-provided structural annotations.
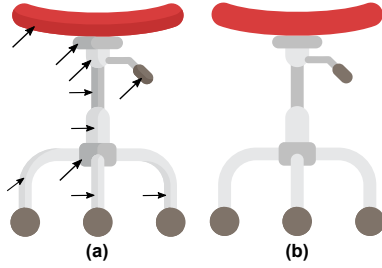


Figure 4: (a) Some closed paths (pointed by arrows) in the clipart represent shading instead of geometry. (b) After removing all the shading path, the remaining closed paths represent geometry only.

proposed on raster image and focus on our application of aiding users to design clipart under unseen viewpoint.

### 3.1. Single-view guiding shape synthesis

Performing 3D reconstruction from single-view input has been a very challenging task. In recent years, with the fast advances of the deep learning methods, the state-of-the-art method has shown the possibility of learning a descriptive latent representation for category-specific objects and synthesizing the corresponding shape represented by voxels, point clouds, or meshes. In this work, we reconstruct the guiding 3D shape $M^G$ based on the input clipart $S$ by following Lun *et al.* [LGK*17]. In their method, they use an encoder-decoder architecture where there are multiple (12 in their paper) decoders that predict geometric data, including (i) the normal maps, (ii) the depth maps, and (iii) foreground probability maps for each viewpoint. For each input sketch, they applied Poisson surface reconstruction [KBH06] on the predicted geometric data to generate the mesh. The entire pipeline is illustrated in Figure 3(a).

We observe several differences between our input clipart and the original sketch data used in [LGK*17], which prevents us from di-

rectly applying their method to our input data. Thus we perform the following preprocesses before we feed the input clipart into their method:

**Shading path removal** One characteristic of the clipart is that it usually describes both *appearance* and *geometry* in the same file. Unlike the 3D model file, which usually describes the geometry only, its appearances are defined through different material or texture files. However, the path elements in the clipart sometimes represent the shading, e.g., reflection (see Figure 4)), which is not directly representing the shape. In this work, we focus on aiding the users to design the geometry part of the clipart, so we categorized path elements in the clipart into two categories, i.e., geometry path and shading path. Furthermore, we remove the shading path before we perform the 3D reconstruction. Currently, this process is performed manually. The main reason is that we do not have lighting information; thus, we can't distinguish path type only based on its curve geometry and color information.

**Color removal** We remove the filled color of each closed path in the input clipart. The remaining curve outlines are served as the sketches' input to the method proposed in [LGK*17].

After the preprocessing, we feed the processed curve outlines into [LGK*17] and obtained the predicted geometric data. Due to the large differences of geometric styles between clipart and the shapes in their training dataset, the resulted point cloud constructed by the predicted geometric data is usually broken, as shown in Figure 5(b).

One possible solution is to re-train their model with a huge clipart dataset; however, collecting large numbers of clipart with multiple viewpoints is expensive and time-consuming. As a result, we choose to use the pre-trained model provided by [LGK*17] as our inference model. Thus, the quality of predicted shapes is usually worse and noisier than the predicted results using the 3D shape sketches that capture the real-world shape style. To further filter out noisy predicted shape data, we use the input clipart to create an auxiliary foreground mask (Figure 5(c)) from the viewpoint of the
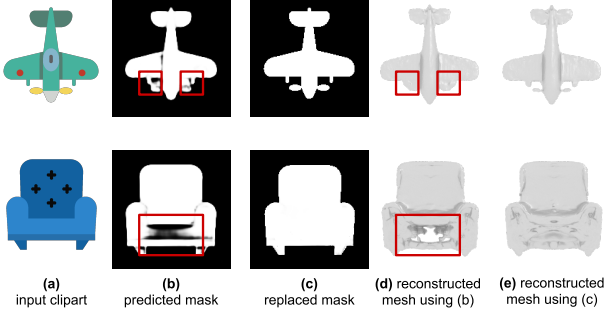
Figure 5: By replacing the predicted noisy foreground probability mask (b) with the known mask derived from the input clipart (c), we can greatly improve the quality of the reconstructed meshes.
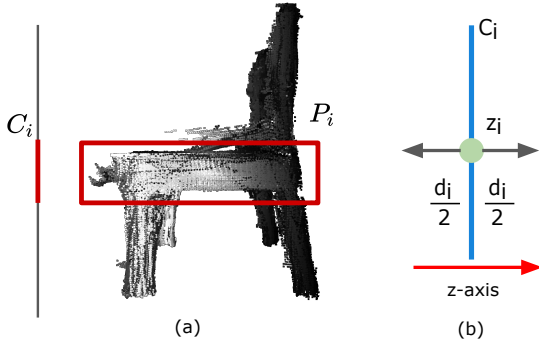


Figure 6: (a) For each curve $C_i$ (the red segment on the left), we project vertices on the guiding shape $M^G$ and obtain a pointset $P_i$ that is enclosed by $C_i$ (inside the red rectangle). (b) Our extrusion parameterization.

input clipart. As we observed, this step turns out pretty helpful for compensating for the bad quality due to the geometric style difference. Please see Figure 5 for reconstructed shapes with and without this auxiliary foreground mask.

### 3.2. User-assisted curve extrusion

After obtaining the guiding shape $M^G$, we want to leverage it to guide the curve extrusion process to make a 2D curve into a 3D volume. Curve extrusion is widely used in designing the CAD model where the designers usually design layout the shape in 2D and extrude certain thickness along the direction into solid volumes. Our goal is to extrude all the closed paths in the input clipart $S$ into volumes and transform them to cover as many vertices in the guiding shape $M^G$ as possible. We assume the extrusion axis is along the z-axis (i.e., the input clipart lying on the xy-plane). And we parameterize the extruded volume $V_i$ of a closed path $C_i$ using the following parameter: (i) the extrude thickness $d_i$, and (ii) the z-coordinate $z_i$ of the centroid of the extruded volume (see Figure 6(b) for illustration).

**User-assisted structural annotation** We allow the users to annotate each closed path in the input clipart with some structural properties. These annotations enable the users to assign the 3D in-
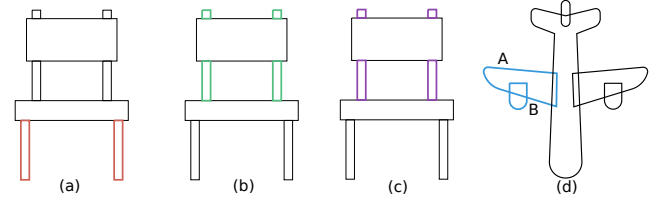


Figure 7: The user can annotate four different types of structural information. (a) The paths of chair legs (in red) are annotated as *multiple objects*. (b) Four paths (in green) are annotated as *same thickness*. (c) Four paths (in purple) are annotated as *same depth*. (d) Path A is annotated as it is in front of path B (blue).

formation they depicted. We provide the following four types of annotations:

- **multiple objects**: for each closed path $C_i$ in 2D, there might be multiple objects in 3D which are occluded. We provide this annotation to the user to assign the number of objects $N$ they think exists. We duplicate the annotated closed path $N$ times before doing the extrusion.
- **same thickness**: for each pair of closed paths $C_i$ and $C_j$, the user can enforce them to have the same extrude thickness (i.e., $d_i = d_j$).
- **same depth**: for each pair of closed paths $C_i$ and $C_j$, the user can enforce the z-coordinate of the centroid ($z_i$ and $z_j$) of the extruded volume to be the same.
- **depth order**: by default, we will leverage the input clipart's layering as the depth orders between closed paths. However, the user can assign the desired depth order to overwrite the default ordering.

Please see Figure 7 for illustrations of each structural annotations.

For each closed path $C_i$, we project the vertices of the guiding shape $M^G$ back onto the xy-plane and obtain a pointset $P_i$ that is enclosed by $C_i$ (see Figure 6(a)). For the closed path annotated as multiple objects, we perform a clustering operation on the enclosed pointset $P_i$ based on the 3D points' positions, e.g., if a closed path is duplicated twice, we generate two pointsets $P_i^0$ and $P_i^1$. And we assign each pointset to each duplicated closed path. And we obtain the best extruded volumes $\mathbf{V} = \{V_1, ..., V_n\}$ by optimizing the following geometry approximation cost function:

$$\text{minimize} \quad E^{cover} + \omega E^{thickness} \tag{1}$$
$$\text{subject to} \quad z_i - z_j = 0 \quad \textbf{same depth} \tag{2}$$
$$z_i - z_j > 0 \quad \textbf{depth order} \tag{3}$$
$$d_i - d_j = 0 \quad \textbf{same thickness} \tag{4}$$

We define the volume coverage cost ($E^{cover}$) as:

$$dist(x, V_i) = \begin{cases} 0, & \text{if } x \text{ is inside } V_i \\ \min_{q \in \Omega(V_i)} \|x - q\|, & \text{otherwise,} \end{cases} \tag{5}$$

$$E_i^{cover} = \sum_{x \in P_i} dist(x, V_i) \tag{6}$$

$$E^{cover} = \sum_{i=1}^{n} E_i^{cover} \tag{7}$$

where $x$ is a point belongs to $P_i$, $n$ is the number of closed paths in input clipart $S$, and $\Omega(V_i)$ represents the surface of the extruded volume. And we define the thickness cost ($E^{thickness}$) as :

$$E^{thickness} = \sum_{i=1}^{n} \|d_i\|^2. \tag{8}$$

The intuition of optimizing Eq. 1 is to encourage the volume to cover as many enclosed points as possible and keep the extrude thickness small. While minimizing the cost function (Eq. 1), the structural annotation is formulated into constraints (Eq. 2 to 4) that enforce the structural relationship between closed paths in the input clipart $S$. The $\omega$ we used in Eq. 4 is set as 100 by doing a grid search over a set of possible values on two verification clipart. We found out the value better balance the scale between both terms and discourage the over-thickness extrusion results.

**Optimization method** To simplify the optimization problem, we propose to use a greedy method to obtain the optimized result. To optimize Eq. 1 for each closed path $C_i$, we first fit an oriented bounding box (OBB) $O_i$ for the enclosed pointset $P_i$. For each $d_i$, we limit the possible values as three side lengths of $O_i$. We initialize $d_i$ as one of the side lengths with the minimum $E^{cover}$ value. And we initialize $z_i$ as the centroid of $P_i$.

Next, we resolve the structural constraints (from Eq. 2 to 4) one by one. For **same depth** constraint, for closed paths $C_i$ and $C_j$ that are annotated as same depth, we move one of their extruded volume (i.e., $V_i$ or $V_j$) that leads to the smaller $E^{cover}$. For **depth order** constraint, we also choose one of the extruded volumes between $V_i$ and $V_j$ and move the z coordinate of the chosen volume (either $z_i$ or $z_j$) that leads to smaller $E^{cover}$. Finally, for **same thickness** constraint, we also choose one of the extruded volumes between $V_i$ and $V_j$ and set their thickness value of the chosen volume (either $z_i$ or $z_j$) that leads to smaller $E^{cover}$. In Figure 8, we show an illustrative example of the optimization process. The main reason for utilizing the greedy method instead of the traditional gradient-based optimization method is that the two sets of variables (i.e., the thickness $d_i$ and the centroid of extruded volume $z_i$) related to each other. Specifically, the different centroid of extruded volume leads to different optimal thickness and vice versa. Although this kind of problem is studied and solvers are proposed for different applications [LZX*08, BDS*12], their convergence is not guaranteed, and it is likely to extend the running time. In our experiment, our greedy method can achieve better results in most of the cases.

## 4. User Interface

Figure 9(a) illustrate our user interface, which consists of two separate areas. The *reference* area shows the input clipart, which acts as the reference viewpoint for designing clipart from the desired viewpoint. The *canvas* area is the place that the user designs the clipart using curve and shape tools. In the *canvas* area, we overlay the visual scaffold under the canvas, and the user can decide if he/she wants to follow the scaffold or not.

In the *canvas* area, we provide two sets of drawing tools, i.e., curve tools and shape tools. In the curve tools, we provide (i) line, (ii) arc, and (iii) freeform tools. The reason for choosing line and arc is because we observed that many clipart could be described solely using lines and arcs. In the shape tools, we provide
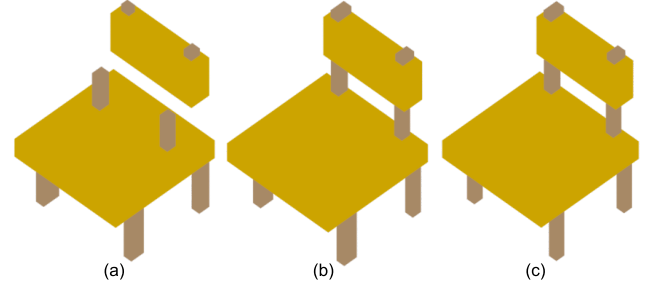


Figure 8: The result after (a) initialization of the thickness ($d$) and the depth value ($z$), (b) resolving depth value, and (c) resolving thickness.
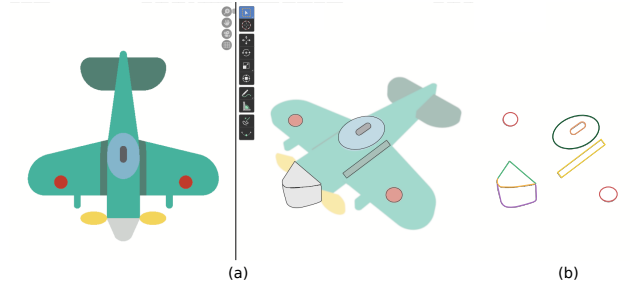


Figure 9: (a) Screenshot of our user interface. On the *reference* area (left), we show the input clipart. On the *canvas* area (right), the visual scaffold is overlaid under the canvas. The user can use curve tools and shape tools to design clipart from the desired viewpoint. (b) In the current design session, the user used different curves and shapes (represented in different colors), including line, arc, bezier freeform tools, rectangle, ellipse/circle, and rounded rectangle. Please see the attached video for the detail of our user interface.

(i) rectangle, (ii) ellipse/circle, and (iii) rounded rectangle. These shape primitives are chosen because they are often used to design clipart from canonical viewpoints. Please see Figure 9(b) for the illustration of different tools.

In the *canvas* area, the user can first sketch out the outline of the shape he/she wants to draw. The sketch can be hidden during the designing process. The user can also draw separate curves and primitives in different layers and move layers forward and backward if he/she wants to.

## 5. Results and Evaluation

We used our user-assisted curve extrusion method to generate several man-made 3D shapes of different clipart. For each input clipart, we compared the 3D shape was generated by our method with the following methods:

**Single-view sketch shape reconstruction** The guiding shape we used in our method was generated by [LGK*17].

**Sketch-based shape retrieval** We perform a sketch-based shape retrieval using input clipart. The goal is to retrieve the most similar shape in ShapeNet [CFG*15] dataset. For each shape in

the ShapeNet dataset, we extracted their outlines using Suggestive Contours [DFRS03] as candidate images. And we extract the strokes of each input clipart as query and compute the similarities between the clipart strokes and contour of each shape using Shape Context descriptor [BMP01].

**Artist creation** We asked an artist to design the clipart from unseen viewpoints given the input clipart, and he generates a reference 3D shape as his reference for each clipart.

We assigned colors for both results using single-view sketch shape reconstruction and shape retrieval from the input clipart. We perform shape registration between the rendered contour image to the input clipart to establish the correspondences. And we project the color obtained by correspondences onto the retrieved 3D shape. For propagating the colors onto all triangle $T$ of the retrieved 3D mesh, we use a formulation of a Markov random field (MRF) problem as follow:

$$E(f) = w_{data} \sum_{t \in T} \mathcal{D}(t, f_t) + w_{smoothness} \sum_{t,s \in \mathcal{N}} \mathcal{S}_{t,s}(t, s, f_t, f_s), \quad (9)$$

where $f$ is the function assigns the color label to each triangle $t \in T$, $f_t$ and $f_s$ are color labels assigned to triangle $t$ and $s$, and $\mathcal{N}$ is the set of all pairs of triangles sharing edges. And we define the data term (i.e., $\mathcal{D}$) as follow:

$$\mathcal{D}(t, f_t) = \begin{cases} \|c(t) - \bar{c}(t)\|_2^2, & \text{if color } \bar{c}(t) \text{ is assigned to triangle } t \\ 0, & \text{otherwise} \end{cases}$$
$$(10)$$

And the smoothness term (i.e., $\mathcal{S}_{t,s}$) is used to measure the spatial consistency of neighboring triangles and is defined as follow:

$$\mathcal{S}(t, s, f_t, f_s) = \begin{cases} 0, & \text{if } f_t = f_s \\ -\log(\theta_{t,s}/\pi)\psi_{t,s}, & \text{otherwise}, \end{cases} \quad (11)$$

where $\theta_{t,s}$ and $\psi_{t,s}$ are the dihedral angle and the centroid distance between $t$ and $s$, respectively. In all the results shown in the paper, we set $w_{data} = 1.0$ and $w_{smoothness} = 10$. And we solved Eq. 9 using the graph cut algorithm [BK04]. Noted that the assigned colors are merely for convenient comparison.

We show all the results of compared methods in Figure 12 and Figure 13. Noted that we only show 3D shapes from upper $45°$ in the main paper, and we show the rendered image from canonical views in the supplemental material. For all shapes, the geometric and appearance styles of our reconstructed results are more consistent with the input clipart compared to the results generated by [LGK*17] and the retrieved results. And compared to the artist's results, it took him on average 1 hour to create the 3D shapes for chair cases and 1.5 hours for airplane cases. Our method automatically produces qualitatively similar results in a fraction of this time (for chair cases, 4 mins including user annotation and extrusion optimization, and 6.5 mins for airplane cases on average).

To further analyze our results, we perform a qualitative evaluation to examine 3D shapes generated by our method compared to the results of other methods. We conduct a comparative user study by showing the input clipart, 3D shapes generated by our method and the alternative methods side-by-side. Then we asked the participants which one better illustrates the underlying geometry. More specifically, we indexed the input clipart as (A) and our method and
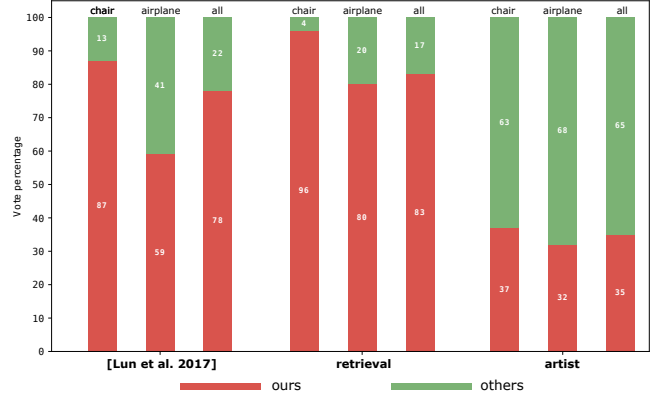


Figure 10: We show the comparative statistics between our method, two alternative methods, and the artist's creation. The result 3D shapes of our method represent the underlying 3D geometry of the input clipart significantly better than other alternative methods (78% v.s. 22% and 83% v.s. 17%). Compared with the artist creation, there are still 35% of votes in favor of our result 3D shapes, which suggests that our method can generate 3D shapes with decent quality.

alternative as (B) or (C). Then, the participant answers: "Which of the two images on the bottom (B) or (C) better represent the underlying 3D structure of the input clipart (A)?". All images are rendered from the upper $45°$. We compared our results to three alternative methods, including *sketch shape reconstruction* [LGK*17], *shape retrieval*, and *artist creation*.

We conducted this survey on Amazon Mechanical Turk (AMT), and the order of all questions was given randomly for each participant. We excluded the response that the answer consistency is lower than 60% of all questions. In the end, 43 out of 55 participants meets our criteria. As shown in Figure 10, 78% of votes prefer our result over that of single-view sketch shape reconstruction [LGK*17] and 83% of votes prefer our result than retrieved result. We compare our result 3D shapes with the artist's creation. In this case, most votes in favor of artist's creation show some room for improvement of our reconstruction method. However, our result 3D shapes still obtain 35% of votes, which suggests that our method can generate 3D shapes with decent quality.

### 5.1. Drawing user study

We have conducted a preliminary user study to evaluate how the visual scaffold can help users design clipart under unseen viewpoints. This study did not aim to evaluate the method of synthesizing the visual scaffold. The participants only use the prepared visual scaffold generated by the authors. The major reason is that to fairly evaluate how the visual scaffold assists the user in designing clipart, we have to use the same visual scaffold for different participants.

**Setup** All tasks were conducted on a 12-inch Surface Pro laptop with a Surface stylus. The study contains three sessions: tutorial, drawing, and interview. The entire study took about two hours per participant.

**Participants** We recruited 12 participants (seven males and five females). All participants are non-professional artists, and only 3 of them have experience in physical-painting or digital-painting. We ask each participant to draw one of the six clipart (three chairs and three airplanes).

**Tutorial session** We designed the tutorial session to help the participant familiarize with the drawing interface we provided and different functions in our drawing interface. There are four stages of the tutorial session that aim to let the user familiarize with functions, including (i) drawing single curves, (ii) connecting curves, (iii) filling color, (iv) how to draw primitives, and (v) the layering concept. For each function, we prepared target results (e.g., rectangles with different sizes or specific layering) and asked them to use the tools to match the targets. In total, we allow the participants to practice in the tutorial session for at most 15 minutes.

**Drawing session** In the drawing session, we show the input clipart to the participants and ask them to design clipart of the same object but from three other viewpoints. For chair clipart, the input viewpoint is from the front viewpoint, and then we ask the user to draw from the top, and the side, and the upper $45°$ viewpoints. For airplane clipart, the input viewpoint is from the top viewpoint, so we ask the user to draw the front, and the side, and the upper $45°$ viewpoints. Each participant design the clipart with and without our visual scaffold in a random order to avoid bias.

**Interview** In the end, we collect feedback from each participant on different aspects of the assistive drawing interface. In the questionnaire, we asked participants about (i) their drawing experience, (ii) their thoughts on difficulties of drawing different viewpoints, and (iii) how they think the visual scaffold aids them during the drawing session. In the last part, we asked the following questions:

- do they think the provided visual scaffold is helpful?
- how the provided visual scaffold affect them while drawing?
- other suggestions or comments on how they think the provided visual scaffold aid can be improved.

Please find the full version of our questionnaire in Section 3 of the supplemental material.

**Drawing result** We show several clipart designed by the participants in Figure 14. Noted that we do not have the exact correct design for all the clipart we asked the participants to design. Thus, in the following discussion, we use the artist's results as the best reference design since the artists are usually more familiar with the workflow and are more likely to design better results. In Figure 14, we can observe that the participant can draw a better ratio between the width and the height of the purple sofa case. The design result with the visual scaffold is closer to the artist's result than the result without the visual scaffold. For the purple sofa case, the participant can design better clipart from the upper $45°$ by mostly followed the provided visual scaffold because the generated 3D shape matched the expected underlying geometry. Simultaneously, even though sometimes the visual scaffold's geometry is not accurate at every location, it can still assist the participants in designing the clipart. For the blue airplane case (shown in the last row in Figure 14), the participant can place the wing at the position closer to the artist's result by designing with the provided visual scaffold. Please see the complete results from all participants in the supplemental material.

### 5.1.1. Interview result and discussion

**Viewpoint Survey** We asked the participants about their thoughts on difficulties of drawing different viewpoints. All of the participants agree that the difficulties vary across viewpoints. Among all participants, eleven out of twelve consider the upper $45°$ viewpoint more challenging to draw than the rest viewpoints. More specifically, for airplane clipart, 83.3% of participants thought the upper $45°$ viewpoint is the hardest to draw, and 16.7% of them thought the side viewpoint is the hardest one. In terms of chair clipart, all participants agree that the upper $45°$ viewpoint is the hardest to draw.

**Scaffold aids survey** Eleven of twelve participants (91.7%) considered the provided visual scaffold is helpful during the drawing process. And they thought the visual scaffold is helpful in the following ways:

- two participants thought the visual scaffold helps them see the overview at the beginning of the design process. It gives them a better idea of how to draw the different layers composing the target object.
- five participants thought the visual scaffold helps them estimate the ratio and scale between different parts of the target object.
- two participants thought the visual scaffold reminds them some details they ignore
- two participants thought the visual scaffold helps them to arrange the layering while designing the target object.

According to the above user feedback, we can observe that our visual scaffold helps design clipart under an unseen viewpoint, especially useful for estimating the ratio and scale between parts. Meanwhile, two participants thought the visual scaffold might interfere with their imagination, thus hindering the design process. The potential reason is that the visual scaffold is sometimes inaccurate, which might conflict with the participants' imagination of the target object.

Overall, most of the participants can design the clipart from the target viewpoint in shorter periods. As shown in Figure 11, the participants saved time on drawing upper $45°$ for both chair and airplane clipart. With the provided visual scaffold, the participants can save 57% of time when drawing complicated parts such as the airplane from the front viewpoint. Meanwhile, we observed that the participants spent more time drawing the chair from the side viewpoint. From the user feedback, we found out that this is because the visual scaffold provides additional details from the side viewpoint compared to their imagination. Even though they spent more time drawing with visual scaffold from the side viewpoint, they can draw more details, leading to better results.

## 6. Conclusion

In this paper, we propose an assistive system that aids the users in designing clipart from unseen viewpoints. The core of our system is a user-assisted curve extrusion method that combines the user-provided structural annotation and a guiding 3D mesh that is synthesized by the single-view shape reconstruction method. We render the generated 3D shape using the same style of input clipart into the visual scaffold. We conducted a user study with 12 users. With our visual scaffold, we found out that the users can design
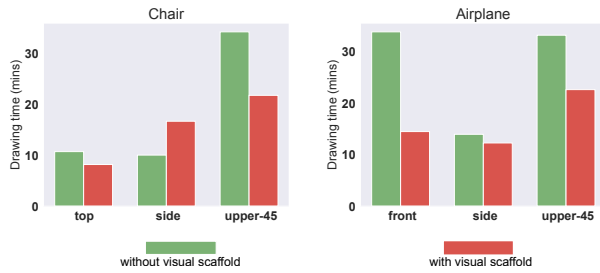
Figure 11: We show the drawing time statistics of the user study. On average, the participants can save 15% of drawing time for chair cases and 40% for airplane cases. The provided visual scaffold is most helpful in drawing the airplane from the front viewpoint (saved 57% of drawing time). And it is most helpful for drawing both the chair and the airplane from the upper $45°$ viewpoint.

better quality clipart from unseen viewpoints using a shorter time than designing without the provided visual scaffold.

There are two parts of limitation in our proposed system, i.e., the 3D reconstruction part, and the user-assisted design part. The limitations of our 3D reconstruction method are twofold. The first part is because we choose to use a pre-trained model of an existing learning-based 3D reconstruction method [LGK*17] to reconstruct the guiding mesh, the quality of predicted shapes is usually not satisfying. Besides the reason mentioned in Section 3.1, another reason is that many clipart shapes are quite distinct from the training data. For example, we can not find similar shapes in the training data for the two airplanes shown in Figure 15; thus, the reconstructed guiding meshes are pretty noisy and broken. Our current method is not able to recover the missing parts in the guiding mesh. The second one is that our extrusion method can not model curvature along the extrusion direction. Since the current extrusion method only extrudes the geometry with a single thickness value. We plan to use different geometry representation to address this issue, such as voxels, bevels, or geometry profiles used in [KFWM17]. In terms of the limitation of the user-assisted design part, we observed that designs clipart from a non-canonical viewpoint (e.g., the upper $45°$ viewpoint) is much harder than a canonical viewpoint. However, our current system does not provide step-by-step assistance for users. In the future, we would love to explore how to design more intuitive tools, e.g., adding block-in guides, or other novel assistance to better aids the users in designing clipart from non-canonical viewpoints.

## 7. Acknowledgement

## References

[Ado20] ADOBE: Adobe illustrator 2020: Image trace, 2020. URL: http://www.adobe.com/. 3

[BDS*12] BOUAZIZ S., DEUSS M., SCHWARTZBURG Y., WEISE T., PAULY M.: Shape-up: Shaping discrete geometry with projections. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1657–1667. 6

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence 26*, 9 (2004), 1124–1137. 7

[BMP01] BELONGIE S., MALIK J., PUZICHA J.: Shape context: A new descriptor for shape matching and object recognition. In *Advances in neural information processing systems* (2001), pp. 831–837. 7

[CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., ET AL.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015). 6

[CGW*14] CHEN H.-T., GROSSMAN T., WEI L.-Y., SCHMIDT R. M., HARTMANN B., FITZMAURICE G., AGRAWALA M.: History assisted view authoring for 3d models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), CHI '14, ACM, pp. 2027–2036. URL: http://doi.acm.org/10.1145/2556288.2557009, doi:10.1145/2556288.2557009. 3

[CSH19] CHEN X., SONG J., HILLIGES O.: Monocular neural image based rendering with continuous view control. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 4090–4100. 2

[CXG*16] CHOY C. B., XU D., GWAK J., CHEN K., SAVARESE S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision* (2016), Springer, pp. 628–644. 2

[DFRS03] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S., SANTELLA A.: Suggestive contours for conveying shape. *ACM Transactions on Graphics (Proc. SIGGRAPH) 22*, 3 (July 2003), 848–855. 7

[DN19] DAI A., NIESSNER M.: Scan2mesh: From unstructured range scans to 3d meshes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 2

[DSTB16] DOSOVITSKIY A., SPRINGENBERG J. T., TATARCHENKO M., BROX T.: Learning to generate chairs, tables and cars with convolutional networks. *IEEE transactions on pattern analysis and machine intelligence 39*, 4 (2016), 692–705. 2

[DTM96] DEBEVEC P. E., TAYLOR C. J., MALIK J.: Modeling and rendering architecture from photographs: A hybrid geometry-and-image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 11–20. 2

[FSG17] FAN H., SU H., GUIBAS L. J.: A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 605–613. 2

[GFK*18] GROUEIX T., FISHER M., KIM V. G., RUSSELL B., AUBRY M.: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2018). 2

[HDS*18] HOSHYARI S., DOMINICI E. A., SHEFFER A., CARR N., WANG Z., CEYLAN D., SHEN I., ET AL.: Perception-driven semi-structured boundary vectorization. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 118. 3

[HLW*17] HENNESSEY J. W., LIU H., WINNEMÖLLER H., DONTCHEVA M., MITRA N. J.: How2sketch: Generating easy-to-follow tutorials for sketching 3d objects. *Symposium on Interactive 3D Graphics and Games* (2017). 3

(a) input clipart    (b) [Lun et al. 2017]    (c) retrieved result    (d) our result    (e) artist's result
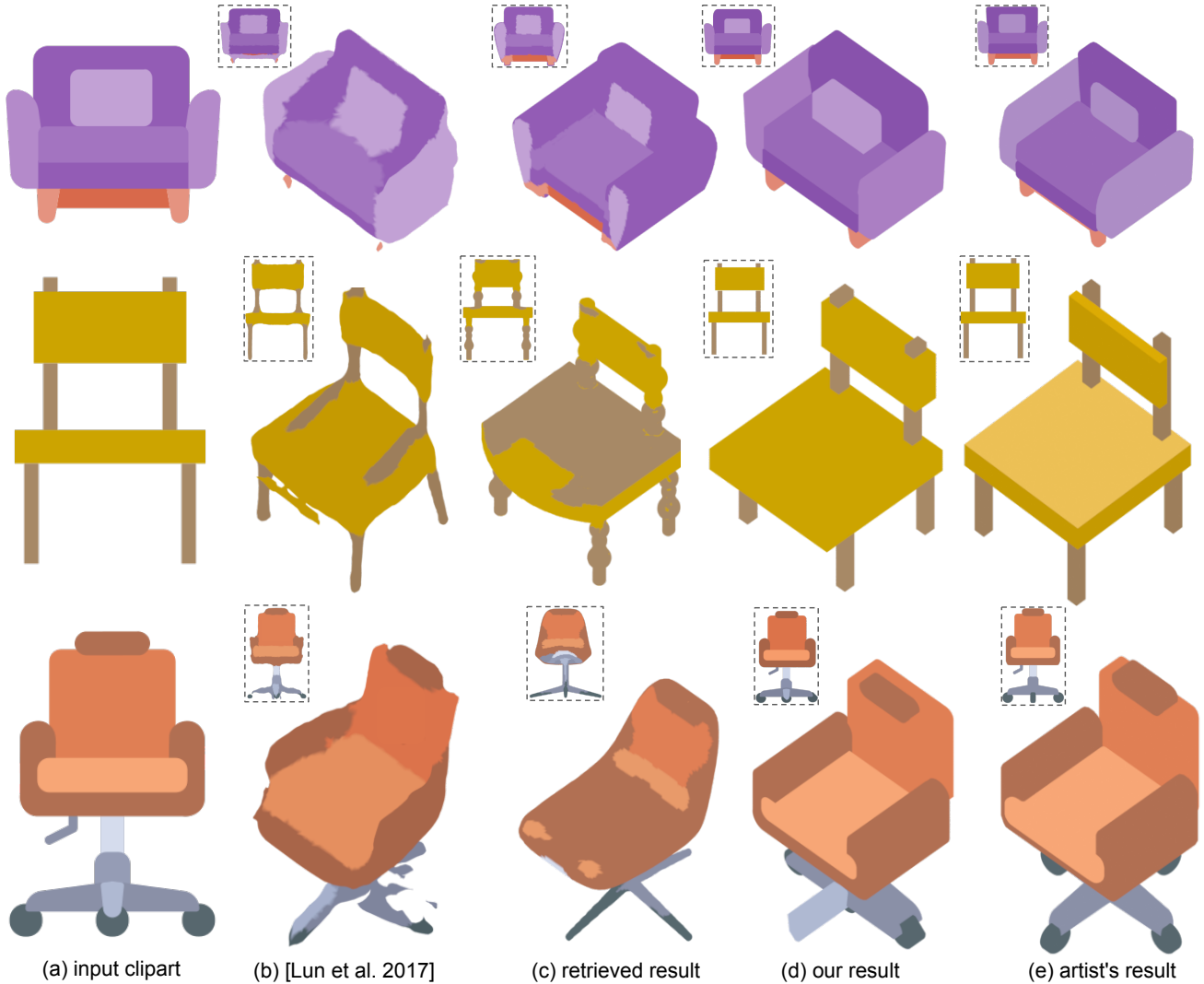
Figure 12: We compared the reconstructed 3D chair shapes using our method with the results generated by [LGK*17], the results using shape retrieval (see the detailed explanation in the main text), and the artist's created 3D chair shapes. Our reconstructed 3D shapes preserve geometric and appearance styles of the input clipart compared to the results generated by other automatic methods. The shape is more regular, and there are few incomplete structures in the reconstructed 3D shape. In the meanwhile, our reconstructed 3D shapes are very close to the artist's created shapes.

[KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2006), SGP '06, Eurographics Association, p. 61–70. 4

[KFWM17] KELLY T., FEMIANI J., WONKA P., MITRA N. J.: Bigsur: large-scale structured urban reconstruction. *ACM Transactions on Graphics 36*, 6 (2017). 9

[KL11] KOPF J., LISCHINSKI D.: Depixelizing pixel art. *ACM Transactions on graphics (TOG) 30*, 4 (2011), 99. 3

[KLA19] KARRAS T., LAINE S., AILA T.: A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 4401–4410. 2

[KLS*13] KOPF J., LANGGUTH F., SCHARSTEIN D., SZELISKI R.,

GOESELE M.: Image-based rendering in the gradient domain. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013) 32*, 6 (2013). 2

[LALR16] LIU Y., AGARWALA A., LU J., RUSINKIEWICZ S.: Data-driven iconification. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)* (May 2016). 3

[LGK*17] LUN Z., GADELHA M., KALOGERAKIS E., MAJI S., WANG R.: 3d shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)* (2017), IEEE, pp. 67–77. 2, 4, 6, 7, 9, 10, 11

[LHES19a] LOPES R. G., HA D., ECK D., SHLENS J.: A learned representation for scalable vector graphics. In *The IEEE International Conference on Computer Vision (ICCV)* (October 2019). 3

[LHES19b] LOPES R. G., HA D., ECK D., SHLENS J.: A learned rep-

(a) input clipart    (b) [Lun et al. 2017]    (c) retrieved result    (d) our result    (e) artist's result
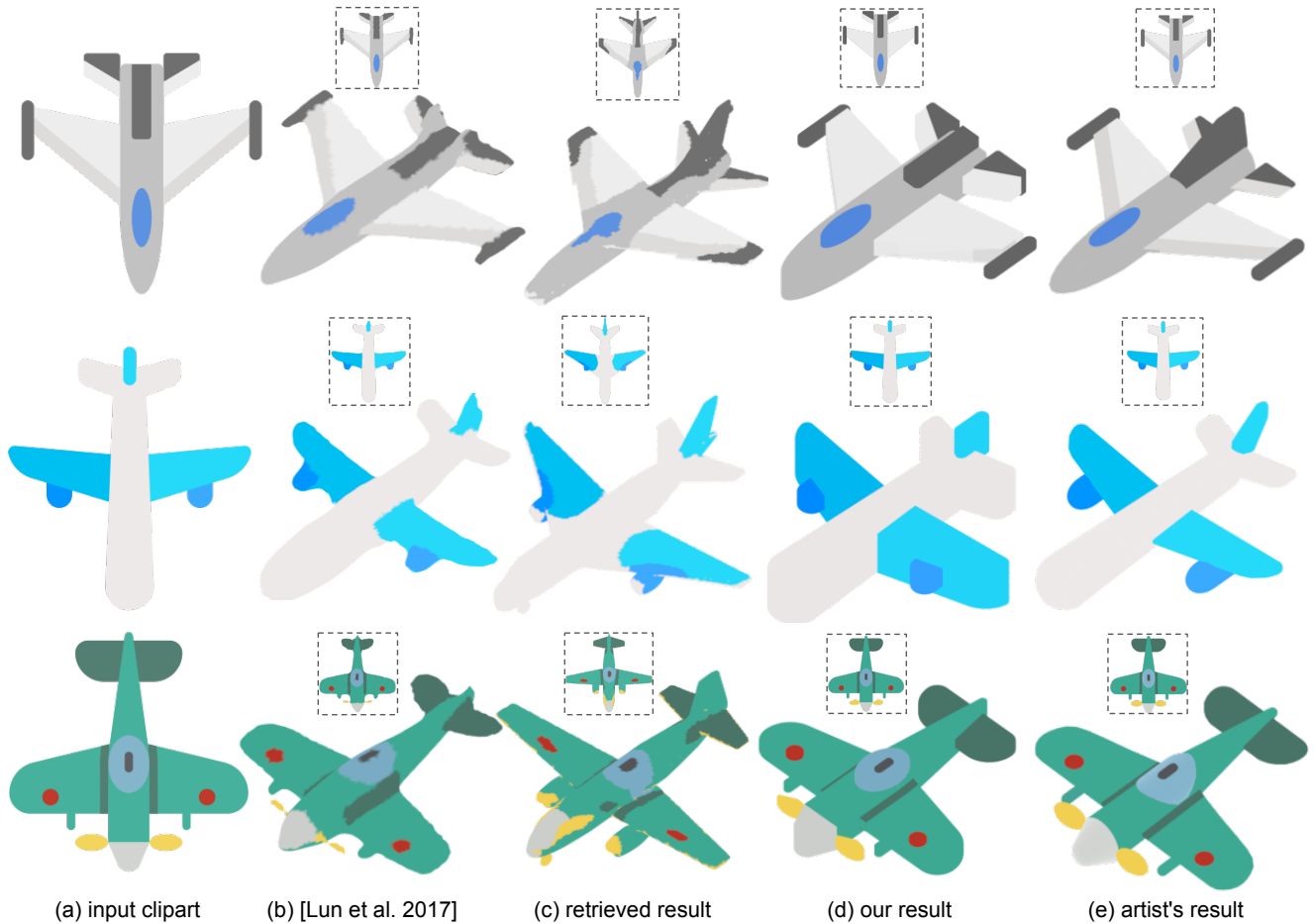
Figure 13: We compared the reconstructed 3D airplane shapes using our method with the results generated by [LGK*17], the results using shape retrieval (see the detailed explanation in the main text), and the artist's created 3D airplane shapes. Our reconstructed 3D shapes preserve geometric and appearance style of the input clipart compared to the results generated by other automatic methods. The shape is more regular, and there are few incomplete structures in the reconstructed 3D shape. In the meanwhile, our reconstructed 3D shapes are very close to the artist's created shapes.

resentation for scalable vector graphics. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 7930–7939. 3

[IJ19]  LIU H.-T. D., JACOBSON A.: Cubic stylization. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–10. 3

[LKS15]  LUN Z., KALOGERAKIS E., SHEFFER A.: Elements of style: Learning perceptual shape style similarity. *ACM Transactions on Graphics 34*, 4 (2015). 3

[LKWS16]  LUN Z., KALOGERAKIS E., WANG R., SHEFFER A.: Functionality preserving shape style transfer. *ACM Transactions on Graphics 35*, 6 (2016). 3

[LLCL19]  LIU S., LI T., CHEN W., LI H.: Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 7708–7717. 2

[LPL*18]  LI C., PAN H., LIU Y., TONG X., SHEFFER A., WANG W.: Robust flow-guided neural prediction for sketch-based freeform surface modeling. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–12. 2

[LYH*15]  LUO S.-J., YUE Y., HUANG C.-K., CHUNG Y.-H., IMAI S., NISHITA T., CHEN B.-Y.: Legolization: Optimizing lego designs. *ACM*

*Transactions on Graphics (Proc. SIGGRAPH Asia 2015) 34*, 6 (2015), 222:1–222:12. 3

[LZC11]  LEE Y. J., ZITNICK C. L., COHEN M. F.: Shadowdraw: Real-time user guidance for freehand drawing. *ACM Trans. Graph. 30*, 4 (July 2011), 27:1–27:10. URL: http://doi.acm.org/10.1145/2010324.1964922, doi:10.1145/2010324.1964922. 3

[LZX*08]  LIU L., ZHANG L., XU Y., GOTSMAN C., GORTLER S. J.: A local/global approach to mesh parameterization. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1495–1504. 6

[MWYG20]  MO K., WANG H., YAN X., GUIBAS L.: PT2PC: Learning to generate 3d point cloud shapes from part tree conditions. *arXiv preprint arXiv:2003.08624* (2020). 2

[MZL*09]  MEHRA R., ZHOU Q., LONG J., SHEFFER A., GOOCH A., MITRA N. J.: Abstraction of man-made shapes. *ACM Transactions on Graphics 28*, 5 (2009), #137, 1–10. 3

[NMOG19]  NIEMEYER M., MESCHEDER L., OECHSLE M., GEIGER A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. *arXiv preprint arXiv:1912.07372* (2019). 2
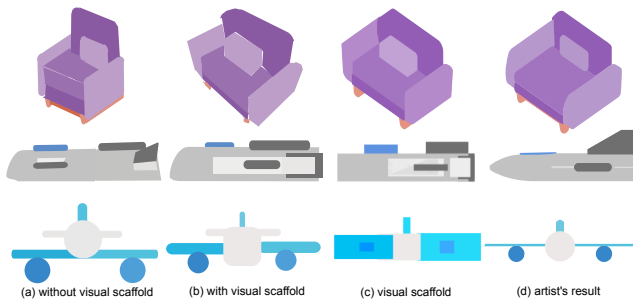
Figure 14: We compare clipart from unseen viewpoints designed by participants (a) with and (a) without our visual scaffold. The visual scaffold is shown in (c). With the aids of the visual scaffold, the participants can (i) capture the ratios between different parts of the target object from unseen viewpoints, and (ii) place different parts in better positions with the aids of our visual scaffolds.
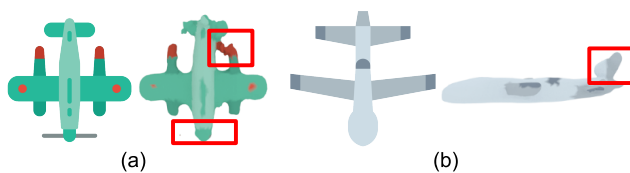


Figure 15: We show two examples of problematic guiding shapes reconstructed using a learning-based 3D reconstruction method. Given the input clipart on the left in both (a) and (b), we can observe the broken reconstruction results highlighted in red rectangles. The reconstruction method fails to reconstruct some thin structures (e.g., propeller in (a)) or add some unnecessary structures (e.g., the vertical stabilizer in (b)).

[NPLT*19] NGUYEN-PHUOC T., LI C., THEIS L., RICHARDT C., YANG Y.-L.: Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 7588–7597. 2

[OTW*19] OLSZEWSKI K., TULYAKOV S., WOODFORD O., LI H., LUO L.: Transformable bottleneck networks. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 7648–7657. 2

[PXW18] PENG M., XING J., WEI L.-Y.: Autocomplete 3d sculpting. *ACM Trans. Graph. 37*, 4 (July 2018), 132:1–132:15. URL: http://doi.acm.org/10.1145/3197517.3201297, doi:10.1145/3197517.3201297. 3

[PYY*17] PARK E., YANG J., YUMER E., CEYLAN D., BERG A. C.: Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the ieee conference on computer vision and pattern recognition* (2017), pp. 3500–3509. 2

[PZ17] PENNER E., ZHANG L.: Soft 3d reconstruction for view synthesis. 2

[SBS19] SMIRNOV D., BESSMELTSEV M., SOLOMON J.: Deep sketch-based modeling of man-made shapes. *CoRR abs/1906.12337* (2019). URL: http://arxiv.org/abs/1906.12337, arXiv:1906.12337. 2

[SCD*06] SEITZ S. M., CURLESS B., DIEBEL J., SCHARSTEIN D., SZELISKI R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)* (2006), vol. 1, IEEE, pp. 519–528. 2

[SHL*18] SUN S.-H., HUH M., LIAO Y.-H., ZHANG N., LIM J. J.: Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *European Conference on Computer Vision* (2018). 2

[SIJ*07] SCHMIDT R., ISENBERG T., JEPP P., SINGH K., WYVILL B.: Sketching, scaffolding, and inking: A visual history for interactive 3d modeling. In *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering* (2007), pp. 23–32. URL: http://www.dgp.toronto.edu/~rms/pubs/ScaffoldingNPAR07.html. 3

[SKSK09] SCHMIDT R., KHAN A., SINGH K., KURTENBACH G.: Analytic drawing of 3d scaffolds. *ACM Transactions on Graphics 28*, 5 (2009). Proceedings of SIGGRAPH ASIA 2009. URL: http://www.dgp.toronto.edu/~rms/pubs/DrawingSGA09.html. 3

[SLHC12] SHEN L.-T., LUO S.-J., HUANG C.-K., CHEN B.-Y.: Sd models: Super-deformed character models. *Computer Graphics Forum 31*, 7 (2012), 2067–2075. (Pacific Graphics 2012 Conference Proceedings). 3

[Ste03] STEPIN M.: Hqx, 2003. URL: http://web.archive.org/web/20070717064839/www.hiend3d.com/hq4x.html. 3

[TDB16] TATARCHENKO M., DOSOVITSKIY A., BROX T.: Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision (ECCV)* (2016). 2

[TZEM17] TULSIANI S., ZHOU T., EFROS A. A., MALIK J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Computer Vision and Pattern Regognition (CVPR)* (2017). 2

[Vec20] VECTOR MAGIC: Cedar lake ventures, 2020. URL: https://vectormagic.com/. 3

[VK78] VANDENBERG S. G., KUSE A. R.: Mental rotations, a group test of three-dimensional spatial visualization. *Perceptual and motor skills 47*, 2 (1978), 599–604. 2

[XCW14] XING J., CHEN H.-T., WEI L.-Y.: Autocomplete painting repetitions. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 172:1–172:11. URL: http://doi.acm.org/10.1145/2661229.2661247, doi:10.1145/2661229.2661247. 3

[XKG*16] XING J., KAZI R. H., GROSSMAN T., WEI L.-Y., STAM J., FITZMAURICE G.: Energy-brushes: Interactive tools for illustrating stylized elemental dynamics. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (New York, NY, USA, 2016), UIST '16, ACM, pp. 755–766. URL: http://doi.acm.org/10.1145/2984511.2984585, doi:10.1145/2984511.2984585. 3

[YK12] YUMER M. E., KARA L. B.: Co-abstraction of shape collections. *ACM Trans. Graph. 31*, 6 (Nov. 2012). URL: https://doi.org/10.1145/2366145.2366185, doi:10.1145/2366145.2366185. 3

[ZTS*16] ZHOU T., TULSIANI S., SUN W., MALIK J., EFROS A. A.: View synthesis by appearance flow. In *European Conference on Computer Vision* (2016). 2